

Trabajando con sesión

Transcripción

[00:00] Hola a todos y todas, bienvenidos y bienvenidas a este nuevo video sobre Servlet. En este video quería poder hacer que cuando nosotros intentemos llegar a nuestra lista de empresas, por ejemplo, nuestra aplicación web nos esté pidiendo sí o sí que hagamos un login.

[00:18] Así como ocurren en Alura, si nosotros queremos ir a `alura.com/dashboard` nos está diciendo automáticamente: “no, vos tenés que loguearte antes”. Entonces, vamos a intentar hacer eso. ¿Cómo es que nosotros lo haríamos? Bueno. Dentro de nuestro login, por ejemplo, cuando nosotros ya sabemos que un usuario existe, lo que nosotros podemos hacer es agregar a ese usuario a nuestra request.

[00:47] Es una cosa que nosotros ya hemos hecho anteriormente, entonces lo que nosotros haríamos sería `request.`, en esta parte, dentro de esta parte del `if` cuando ya vimos que el usuario existe: `request.setAttributes`, y acá como nombre del parámetro podríamos pasar, por ejemplo “LoginUsuario”, a ver si lo escribí bien, sí, “LoginUsuario”. Bien.

[01:13] Y enviaríamos, por ejemplo, a nuestro usuario, esa sería una forma que podríamos intentar hacer. ¿Y a quién se lo enviamos a ese atributo usuario? Bueno, en este caso, por ejemplo, me gustaría enviárselo acá en ese `redirect` que nosotros vamos a nuestra `listaEmpresas`. Entonces me gustaría mostrar en nuestra `listaEmpresas` que ese usuario fue el que se lo logó.

[01:42] Entonces vamos a ir a nuestro JSP. Antes voy a guardar, "Ctrl + S", acá en mi login. Y en nuestro JSP de listaEmpresas voy a colocar por acá un usuario Logeado: y vamos a usar una expression language para poder obtener ese usuario que nosotros hemos enviado acá. Entonces nosotros usamos el nombre de loginUsuario como llave para poder obtener ese objeto.

[02:05] Entonces vamos a usar eso, loginUsuario. LoginUsuario. y acá login, ese usuario que hemos enviado es del tipo usuario y ese tipo usuario tiene un string llamado login que nosotros lo podemos obtener con un getLogin, y nuestro Expression Language es inteligente lo suficiente como para saber que tenemos que usar el método getLogin.

[02:31] Acá solo para dar unos espacios, para dejarlo un poco más bonito, voy a colocar unos br solo para dar unos espacios en nuestro JSP. Y vamos a intentar, vamos a ver qué es lo que ocurre. Voy a dar un run en el servidor y vamos a ver en nuestro loginForm, vamos a colocar Bruno 12345, enviar, vamos a ver. Tenemos un usuario logado, pero no aparece nada. ¿Cómo que no aparece nada?

[03:06] Si yo puse acá en nuestro JSP que tenía que obtener el login y nuestro login también estaba haciendo un setAttribute en nuestro request. ¿Por qué? ¿Por qué no está apareciendo? ¿ueno, ¿qué es lo que ocurre? Este usuario que nosotros hemos colocado en nuestro request dura solamente en cuanto está viva, digamos entre comillas la request.

[03:32] O sea en el tiempo que dure esa request, ese usuario puede ser obtenido. Ahora nosotros estamos haciendo un redirect acá, ¿qué es lo que significa? Significa que el navegador va a hacer otra request para poder ir hasta listaEmpresa. Entonces voy a mostrarles acá en un gráfico que hice que ¿cuál es nuestra situación?

[03:54] Nosotros tenemos esta situación, una request, llegamos en el login, colocamos dentro de esa request nuestro usuario y enviamos de nuevo al

navegador la respuesta. Bien. Y el navegador tiene que hacer un redirect y esa redirect crea una nueva request y va a listaEmpresas, pero esas nuevas request no tiene los datos que tenía la request anterior.

[04:21] Entonces, nuestro usuario que colocamos acá ya no existe acá. ¿Qué es lo que ocurre entonces? Esto para hacer una analogía es como cuando nosotros vamos al banco, por ejemplo. Cuando nosotros llamamos al banco, hacemos una llamada por teléfono y el banco, ¿qué es lo que nos pregunta? Básicamente nos pregunta todos los datos nuestros.

[04:42] Nos pregunta, quiénes somos, nombre y apellido, un documento, nos está pidiendo tal vez si es sobre una tarjeta de crédito, nos está diciendo cuáles son los últimos cuatro números de su tarjeta de crédito. O sea, nos está pidiendo muchos datos para poder saber que somos nosotros. ¿Y qué es lo que ocurre si nuestra llamada cae? Si es por alguna razón, falla nuestra llamada y tenemos que volver a llamar al banco. ¿Qué es lo que ocurre?

[05:15] Nos piden de nuevo todos los datos, nos piden nombre y apellido, documento, etcétera. De nuevo para comprobar quiénes somos. Esa sería más o menos la situación en la que estamos ahora en este momento. Pero eso no me es útil. No es útil, porque yo necesito saber quién es ese usuario y mantener esos datos durante todos los procesos que él va a hacer ese usuario.

[05:37] ¿Entonces, cómo podemos resolver esto? Voy a aplicar otra analogía para resolver ese problema. Cuando nosotros vamos a una institución del gobierno, por ejemplo, para hacer algún trámite, nosotros llegamos y vemos un tótem. En ese tótem colocamos nuestro usuario, nuestro número de documento y obtenemos un papel con una contraseña, con un número, un id.

[06:08] Ese id sirve para identificarnos dentro de ese proceso. Dentro de ese trámite sirve para identificarnos. Entonces nos llaman a través de ese número, nos dicen “puede ir para esta otra caja, por ejemplo, que es donde ahí te van a atender con ese número”. Entonces de un lugar para el otro nos llevan con ese

mismo número y con él nos pueden identificar. Entonces más o menos eso es lo que yo quiero lograr.

[06:29] Ahora, ¿cómo hacemos eso con lo que nosotros tenemos en nuestro servidor? Tomcat nos disponibiliza una cosa llamada session ID. Esa session ID es un número hexadecimal largo, aleatorio también que se crea en el momento en el que un navegador desconocido o un aparato desconocido hace una request por primera vez a nuestro servidor.

[06:59] Cuando hace esa request nuestro Tomcat dice: “No conozco a esta persona, no conozco a este navegador, voy a crear un session id, un número y cada vez que esa persona, ese navegador me envíe una request yo voy a recibir ese número y se lo voy a enviar de nuevo”.

[07:15] Entonces acá Tomcat crea sesión id, se lo envía a nuestro HTML, a nuestro navegador y nuestro navegador en el redirect también nos trae de nuevo ese session id para decir: “Hey, yo soy el que antes te hizo esa request, ¿te acordás?” Entonces nuestro Tomcat con ese session id dice: “sí, me acuerdo de vos”.

[07:35] Entonces ya tiene un mapa de lo que ha hecho esa persona anteriormente, ¿cómo es que conoce ese mapa? A través de un objeto llamado HTTP session. Ese objeto está dentro de nuestro servidor y nuestro Tomcat crea un mapa de session id que es nuestra llave y un HTTP session que es nuestro valor.

[03:02] Entonces están vinculados, la llave puede hacer que puedas acceder a nuestro HTTP session, que tiene datos. Entonces vamos a ver esto en el navegador. Si nosotros vamos acá a ver, voy a apretar F12, vamos a ver dentro de la pestañita de network, voy a dar un F5 acá. Hacemos clic en nuestra entrada, en esta petición esta request que hicimos y vean que tenemos una pestañita de cookies.

[08:36] Puede ser que ustedes vean esta pestañita, pero si van a cookies van a ver esto, y acá pueden ver ese JSESSIONID. La J viene de Java. Entonces, Java session id, bien. Ese JSESSIONID es creado automáticamente cuando yo hago esa request y por ejemplo, si yo hago un modificar y hago clic en enviar, vean que acá yo tengo una segunda request. Es esa entrada.

[09:09] Vean que yo pude ver que es un redirect. Entonces esa es la segunda request que enviamos y vean que es cookie es igual a nuestro cookie anterior. Son los dos cookies iguales. ¿Entonces, qué es lo que está diciendo esto? Nosotros generamos un cookie al comienzo y lo enviamos de nuevo en la segunda request.

[09:31] Que es eso que les estaba mostrando acá. Tenemos una session id y lo enviamos de nuevo en la segunda request. Vamos a ver, en el caso en el que tenemos un navegador que es diferente a este. Vamos a abrir una nueva ventana de incógnito dentro de Google Chrome, "Ctrl + Shift + N" en una ventana de incognito. Podemos ir de nuevo a nuestro localhost/gerenciador/entrada, listaEmpresas y con F12 si yo hago clic en modificar y enviar, vemos que esta cookie es diferente a esta cookie de acá.

[10:13] Miren, esta empieza con 2EC800 y esta, con 6B23. Son diferentes. ¿Por qué? Porque cuando abrimos una ventana de incógnito es considerado un navegador diferente. Si yo abro Microsoft Edge va a ser un navegador diferente y va a tener una cookie diferente, un session id diferente. Entonces, con esto podemos ver que esa session depende exclusivamente de ese navegador en ese momento.

[10:43] Entonces lo que nos queda es algo parecido. Con ese session id nosotros lo creamos en la primera vez que llamamos a nuestro login a cualquiera en realidad, con cualquier petición que se haga, se crea ese session id, lo devolvemos a nuestro HTML, a nuestro navegador y guardamos un HTTP session dentro de nuestro servidor.

[11:09] Con ese HTTP session nosotros podemos guardar datos acá. Entonces vamos a utilizar eso a nuestro favor, ya que tenemos un objeto que se mantiene a lo largo de toda la sesión de todas las requests que ese navegador hace. Entonces vamos a aprovechar eso. Vamos a ir a nuestro código y acá en vez de colocar un usuario dentro de nuestra request, nosotros vamos a obtener un HTTP session.

[11:41] ¿Cómo obtenemos eso? Vamos a colocar HttpSession session = request. Esa sesión nosotros la podemos obtener a través de la request. ¿Por qué de la request y no de la response? Así fue decidido cuando se creó esto de la sesión. Entonces simplemente obtenemos de la request.getSession. Ya nos está dando automáticamente esa opción.

[12:14] Y entonces en vez de usar, de colocar nuestro usuario en la request, que como dijimos, la request muere cuando realizamos otra request, esos datos mueren, nosotros vamos a colocarlo en nuestra sesión, sí, lo escribí bien. "Ctrl + S", bien, entonces vamos a colocar ese atributo en nuestra sesión. Entonces lo colocamos en nuestro acá HttpSession.

[12:42] Entonces, cada vez que la persona intente entrar en una acción enviando esa sesión automáticamente va a poder acceder a ese objeto que yo guardé dentro de mi HttpSession, entonces vamos a ver qué más falta. Dentro de nuestra listaEmpresa podemos decir: "¿Cómo vamos a hacer para modificar esto para que nuestro Expression Language consiga obtener ese objeto?"

[13:13] No necesitamos hacer nada. ¿Por qué? Este expression language es lo suficientemente inteligente como para saber que si no encuentra ese dato en la request, entonces va a buscar ese dato en la session o en la sesión. Entonces no necesitamos hacer ninguna modificación acá. Vamos a parar el servidor e iniciarlo de nuevo, solo para asegurarnos que se limpia todas las cookies.

[13:43] Por cierto, una cosa que no expliqué. Esta cookie es solamente un archivo de texto con un texto que es generado y se lo enviamos al navegador,

pero no es nada más que un archivo. Bien, entonces recuerden este número F1B. Si nosotros damos un F5 acá, hacemos clic acá, van a ver que en este caso aparecen dos sessions, dos requests. Vamos a hacer lo siguiente.

[14:15] Voy a cerrar esto y voy a enviar ahora sí una entrada ListaEmpresas. Si yo hago F12, en vez de lista empresa, vamos a ir a nuestro login, LoginForm. Tenemos acá que ven que finaliza con un número diferente del anterior, porque nosotros paramos el servidor y lo hicimos iniciar de nuevo.

[14:42] Entonces, ahora si yo intento loguearme, Bruno, 12345, si puse bien el número van a ver que nuestra sesión es igual y ahora sí nuestro usuario aparece como Bruno. Entonces podemos ver que estamos compartiendo realmente esos datos. ¿Por qué? Porque hemos guardado el dato del usuario en nuestro HttpSession, que es parte de nuestro servidor Tomcat, lo hemos guardado acá y usamos solamente la llave para poder acceder a él.

[15:18] Entonces, si nosotros, por ejemplo, vamos a Firefox, y vamos a la misma URL, localhost:8080 entrada y acá vamos a LoginForm, es con L mayúscula, LoginForm, vamos a apretar F12 para acceder acá. Y vamos a ir a nuestra red. Si yo hago acá Bruno 12345, ver consulta, no guardar, vemos que dentro de nuestros datos tenemos nuestras cookies y acá tenemos una JSESSIONID diferente a, por ejemplo, está de acá, 647 termina este, y este es 029.

[16:11] Entonces con eso hemos resuelto ese problema de poder recordar al usuario a lo largo de la sesión. Ahora nos falta el problema de que nosotros tenemos que evitar que la persona consiga entrar en listaEmpresas sin antes haberse logueado. Para eso lo vamos a ver en el próximo video, nos vemos